

Synthesizing Carnatic Music with a Computer

M. SUBRAMANIAN

The term 'computer music' is generally applied to producing music from notation or data using a Computer Sound Card installed in a computer or a synthesizer. It thus implies that the music is synthesized or created artificially approximating as closely as possible the tones of musical instruments. Although the advent of multimedia (simultaneous use of text, pictures and sound on a computer) has led to the publication of a large number of CD titles relating to music, these mostly have music recorded from a performer (though occasionally there may be some synthetic music in such titles), and are not considered as generating 'computer music'. Again, artificial music produced using analog devices are not considered as computer music, the essential requirement being that the music is generated from digital data. This article describes the present situation in synthesizing Carnatic music with the computer, the problems and possible solutions.

Western musicians and composers have been extensively using the computer in the field of music for the past decade or so. The synthesizer in the computer, similar to the synthesizer of electronic keyboard instruments, can play simultaneously more than one 'voice', i.e., more than one instrument (melodic or percussion) playing its own notes. Computer music greatly assists composers of Western music, with its emphasis on orchestration and harmony. A composer can immediately listen to his ideas without waiting for it to be played by an orchestra. In India, composers of film music, which has a high content of orchestration, have been using computers for synthesizing their music for a considerable time now. "It is simply amazing what a PC can do to music today. One can practically have a studio at home." (Sandeep Kanjilal quoted in 'Computers and Music', *PC Quest*, June 1992, p.16). A large number of software (some shareware) in a wide range of prices are available for composing music using the staff notation.

"Whereas in the west where billions of dollars are spent in researching for computers in the field of music, India is way behind. The fact that traditional Indian classical music can never be replaced by computers is unquestionable". ('Computers and Music', *PC Quest*, June, 1992, p.17-18). This article, written more than seven years ago, also mentions the names of Louis Banks, Loy Mendonza, Vanraj Bhatia, Ilayaraja and Viju Shah as composers using computers, and goes on to say that "in the field of Indian classical music someone still has to make the first move", which was not quite true since the same issue of *PC Quest* also mentions the efforts made by T. Narayanamurthy (then Deputy Director General in the Department of Telecommunications) who made the first attempts to generate Carnatic music using computers. In fact, he had designed his own operating system and even hardware for the work at a time when the personal computer had not become popular. However, he did not bring out a software package for distribution and his efforts remained in the realm of experimentation, though he gave demonstrations at the Music Academy, Madras.

Carnatic Music and the Computer — Problems

Reasons for the reluctance to accept the computer as a tool to generate Carnatic music are not far to seek. The factors that make it very difficult to synthesize Carnatic music on a computer are as follows:

(a) Although Carnatic music has a large number of full-fledged compositions published with notation, the notation system is only skeletal. If one keys in the notes alone on a harmonium or an electronic keyboard, in most cases the music would be totally unrecognizable. The notation system does not show the nuances or *gamakams* which are essential to get the *raga-bhavam* in most of the ragas.

(b) Even though Sri Subbarama Dikshithar in his magnum opus *Sangitha Sampradaya Pradarsini* provided symbols for the various *gamakams* used nearly a century ago, and though more recent publications by C. S. Ayyar and S. Vidhya use gamakam symbols, these are qualitative descriptions and are inadequate for generating music on the computer which would need a precise quantitative description of the gamakam.

(c) Carnatic music is mainly melodic. A concert may have only a single accompaniment for the main artist (a vocalist or instrumentalist like a Gottuvadyam player). Both artists play the same melody unlike in Western music, where different instruments may be playing different scores. The computer thus does not greatly help a composer who can easily 'hear' his melody in his mind.

(d) The same composition with the same notation may be sung by artists of different schools in widely different ways, all acceptable within the raga's framework.

(e) Extemporization is a basic feature of Carnatic music. Although it is the *alapana*-and *kalpana-swaranam* which are entirely extempore, artists do resort to extemporization even in compositions, and this is often welcomed by listeners.

(f) The score played by the percussion accompanist is varied and personalized. Computerized percussion is often felt to be monotonous by a Carnatic music listener.

Thus on the one hand there are many difficulties in synthesizing Carnatic music, and on the other hand the motivation for synthetic music is also lacking.

Programs for Synthetic Carnatic Music

In spite of these difficulties, I had tried to generate Carnatic music on the computer with gamakams closely approximating the natural singing. The motivation was to produce a program which could explain the Carnatic music system (theory) with text, visuals and audio (while a book can only give a textual description). The program was released in 1994 for public use. It explained the Carnatic music system in five modules, one of which (the *ragam* module) explained eighty ragas in detail. All the modules had audio, visuals, or audio-visuals, which could be accessed while reading the text. Complicated *gamakams* used in ragas like Begada, Arabhi, Devagandhari, Surati, etc., were reproduced reasonably accurately. As the use of sound cards in computers was then not popular (and they were also expensive), the music was produced through the personal computer's speaker, with no control on the

tonal quality or volume. Nevertheless the software was appreciated and well received, indicating that there was need for such a program. This was followed by another program which contained beginner's lessons but which could be in a wide range of *adhara srutis*, enabling practice by the student in his or her own normal sruti. Subsequently, in 1998, I released two programs which ran under the Windows operating system and produced synthetic music through the sound card in the tones of the Veena and flute. The first program was basically the earlier DOS-based program but producing music in the tones of the Veena and flute, while the second one was entirely new. It enabled generation of synthetic music by entering notation in the traditional *sa-ri-ga-ma* style. With the advent of CD Rom drives, a program to explain the Carnatic music system can be produced on a CD Rom with recorded music including vocal music, but synthetic-music programs have the advantage of being played from the hard disk with quicker access to the different modules; they also enable generation of the same music on different instruments and srutis. A program to generate music from notation, of course, cannot be developed without synthesizing the music. In the following pages, I propose to explain the basic principles used in generating synthetic Carnatic music through a sound card, and the problems faced.

Computer and Sound — Midi and Wave Devices

The common sound card found in most personal computers has basically two different sets of chips for producing music. One set — the 'midi synthesizer' — has a chip similar to the synthesizer chips in electronic keyboards. It produces the tones of different instruments generally using a technique called frequency modulation synthesis. It can take data from files which usually have an extension 'mid' and generate the note of required pitch on the specified instrument for the duration specified. Usually four melodic instruments and two percussion instruments can be simultaneously played. Some sound cards provide for more instruments. For better quality, some more expensive sound cards produce synthetic sound using wave-form samples of instruments. The midi device cannot produce vocal music, though some sound effects like clapping are included.

The second set of chips of the sound card consists of the digital analog convertor and analog digital converter (together forming the wave device). These can be used to record all varieties of sound including vocal music and to replay them. However, unlike the cassette tape, the sound is recorded digitally, i.e., the sound is sampled many thousand times in a second (44,100 times for the best quality, but it could be less — 22,050 or 11,025 times a second), and the value of the amplitude of the sound is recorded as a number which may vary between -32768 and +32767 (for a 16-bit recording). This is done by the analog-to-digital converter. The digital-to-analog converter reproduces the music by reading these numbers and generating electric current in proportion to the number, which is fed into the amplifier and speaker. Files generated by this process tend to be very large. (At a sampling rate of 11,025, one minute of recording will occupy 1,323,000 bytes, whereas one minute of a simple midi file may be around 5000 bytes.)

Computerized synthetic music generally uses the midi device, which is fast and generates music from relatively small files. The music composing programs (which use the staff notation) convert the information from staff notation into commands recognizable by the midi device.

Thus it is the midi device which is more convenient for generating synthetic music.

However, Western music does not generally require minute changes in pitch (microtonal variations) or smooth movements between and around notes as required in Carnatic music; some forms of Western music such as jazz require such movements to a limited extent. The midi device provides a command called 'pitch bend' enabling changing the pitch of a note from its normal position. Unfortunately, the extent of pitch variation for a given number in the command has not been standardized. Most sound cards provide for a variation of one major tone (about $9/8$) up or down which is not adequate for all kinds of gamakams used in Carnatic music; the latter sometimes requires smooth movements over a range of $4/3$ (a fourth). Further, the standard list of instruments available with most midi drivers does not include any South Indian instrument. (Sitar is provided under the 'ethnic' category, but the quality of the sound is not always satisfactory.) These problems made it necessary for me to look for some other way of synthesizing Carnatic music.

Wave Device for Synthetic Carnatic Music

The wave device, though primarily meant for recording, is more flexible in the sense that it can play any sound. To use this to produce synthetic music, it is necessary to calculate and create the values that would be generated if the music has been recorded — i.e., 11,025 numbers per second have to be calculated for the lowest sampling rate. To calculate these numbers, the information required would be (a) the pitch of the sound; (b) the pattern of the wave form of the particular instrument at that particular pitch; and (c) the amplitude of the wave at that point. The second parameter is easily ascertained by recording the required instrument's tone at different pitches and analysing them using Fourier analysis, or by actually using the sampled wave for one period. The calculations are a bit complex and would consume a lot of time with slow computers since 11,025 values are to be calculated for one second of music. (11,025 is the lowest possible sampling rate; for better quality, higher rates — 22,050 or 44,100 — have to be used.)

The synthesis presents no difficulty when notes are sounded separately, as for two octaves only about 24 wave-form samples would be required. But in the case of most instruments, particularly the South Indian Veena, the composition of overtones may vary widely over the pitch ranges. In general, the higher pitches have less overtones, but other factors such as the formant of the instrument affect the exact composition of overtones. A musical phrase with a transition from (say) Gandharam to Dhaivatam would traverse through the entire pitch range with varying overtone composition. While the synthesized wave would sound good for separate notes, it will sound metallic or hollow when there is such a large-scale smooth transition.

The second problem is the time taken for calculating the samples. To be useful, the music has to start immediately when a command (by clicking the mouse) is issued. One solution is to create the samples for, say, four seconds, start playing the music, and while the music is played create the next block of four seconds. This is a well-known technique (called double buffering) used in processing sound in computers (recording or playing). The main requirement for this technique to work is that the time taken to synthesize the music should be much less than the time taken to play it. With the algorithm developed by me for calculation,

this criterion was satisfied in PCs with Intel 486 processors working at 66 MHz and faster computers. In fact, with the 486 processor, the time taken to synthesize the music was about 40 per cent of the playing time, and with Pentiums it was far less.

However, the problem of wave-form definition for transitions remained. An approximation was used and a small number of wave forms were defined for different ranges, specially for transitions. The approximation was not only in respect of ranges but also the manner in which the pitch varied. This was the greatest problem of all, since the music may stay at a pitch for some time and then start moving up or down. It may also start moving immediately after the start. To provide for all possibilities, a very large number of wave-form definitions would be required, which could put a heavy strain on the memory resources of the computer. The final product has still some unsatisfactory tones for the Veena, where the pitch varies over a large range. In the case of the flute, the overtones are quite limited and therefore the usual problem of tonal quality during pitch variations did not come up.

The principles used could be extended to other instruments. But for instruments like the violin which has a large number of overtones and whose fundamental pitch for a given sruti (*kattai*) is double that of the Veena, a sampling rate of 11,025 would not be adequate. The sampling rate decides the highest possible frequency that could be recorded (Nyquist limit). Instruments with higher pitches and larger numbers of overtones need a higher sampling rate. For such instruments, a larger number of calculations per second would be required, and thus faster computers.

Gamakams on the Computer

The actual production of Carnatic music on the computer, with all its nuances, required developing a suitable notation system. For the program which only explained the Carnatic music system, where the user is not required to key in any data (except for some quizzes), I developed a system which directly used the frequencies and durations. Initially an attempt was made to develop 'C' language functions for traditionally defined gamakams like *kampitham* (shaking a note), *thirupam* (reaching a higher note and then coming to the lower note in *arohanam*), *jaru* (smooth slides), *janta* (*sphuritam* and *pratyaghatam* — stressing the second note of a pair). Except in the case of *jaru*, these were not found convenient. Instead, the entire music was conceived in phrases, and each individual phrase was defined by (a) the total number of notes in the phrase (including *anuswarams*); (b) the individual frequency and duration of the first note; (c) the duration of transition to the next note; (d) the duration of the next note, and so on. Suitable codes were developed to define these parameters, as also to indicate periods of silence.

Thus the Rishabham of Carnatic Saveri was coded as 155 0 166 20 5 172 5 10 166 15 5 170 5 10 166 10 150 5, where the first number indicated that there were 5 notes in the phrase (the second number 0 being for internal use with reference to the wave-form definition); the number 166 represented the frequency of Shadjam (approximately 3 *kattai*, i.e., the third white note on the keyboard); the next number 20 indicated the duration of Shadjam; the next, 5, the duration of transit to the higher note of frequency, 172, and so on. The last note is Shadjam for a duration of 9 units. The durations are in 10 milliseconds (one-hundredth of a second). The number 150 indicated silence for the duration of the next number. This coding system

was found to be the fastest for calculation: the code was stored in binary form for reading and calculation, which speeded up the process. It took considerable experimentation and practice to get the correct coding for different gamakams, and my four decades of Veena-playing helped me in trying to quantitatively define the gamakams. It may be noted that the Rishabham frequency is less than $166 \times 16/15 = 177$, and even less than $166 \times 256/243 = 174$.

As I did not have the tools to analyse live music gamakams at that time, and as the style of singing gamakams varies among artists, I took the route of experimenting and generating the gamakam as close as possible to the form in which it was taught to me.

Some Observations

The experience of generating gamakams synthetically led me to a few interesting discoveries:

(a) Kampitam, which is a single classification, can have many ramifications, the two basic types being those anchored on the lower note (more common, as in Saveri Rishabham), and those anchored on the higher note (as in Surati Nishadam).

(b) When Kampitam is anchored on the lower note, the duration of the lower note is much more than that of the higher end of the gamakam. The upward transition time is generally less than the downward transit time.

(c) The frequency of the upper end of the note in the case of 'black notes' like the Carnatic Suddha Rishabham is not critical. Often the same musical experience is obtained with different frequencies by adjusting the transit durations and the duration of stay in the upper note. It would appear the brain perceives an average value for the note which could be a weighted average of the frequencies in the entire phrase, giving due weightage for the durations. This is a subject which would require further study in the realm of musical cognizance.

(d) In the case of Carnatic Suddha Rishabham, Sadharana Gandharam, Suddha Dhaivatam and Kaisiki Nishadam, generally the lower end of the gamakam is the note below (Chatusruti Rishabham, etc.), but by using a frequency slightly higher than the lower note, the main note is itself felt to be higher in pitch (for the Suddha Rishabham of ragas like Pantuvarali, the Sadharana Gandharam of Ranjani, etc.)

(e) The durations of notes and their transition times are very important. The same phrase with the same frequencies but with different durations does not generate the required gamakam.

(f) A gamakam within the note (as in the Gandharam of Kalyani) requires an oscillation over a range of a relative frequency of about 41/40. Here again the frequency of the upper end of the gamakam could be changed, and with change in duration an acceptable gamakam could be produced.

(g) In the many cases of gamakams on Suddha Madhyamam (in Sankarabharanam) and Kaisiki Nishadam (Bhairavi), the lower end of the gamakam is below the main note and the upper end at Panchamam and Shadjam respectively (or slightly lower). If the lower end was placed at the note itself, the raga-bhavam changed drastically.

(h) Only janta gamakam could be standardized. It is generally produced by touching the lower note momentarily before the second note of the pair is sung. A duration of 3/100 to 5/

100 per second for the lower note and an equal transit time was found adequate. With this *akara*, *janta* could be produced reasonably well.

(i) *Jaru* was also easy to code. It was found that the duration of transit was critical for *raga-bhavam* in certain cases. For instance, the phrase 'sa ni pa' of *Kedaram* and *Neelambari* differed considerably in the duration of descent.

(j) *Kakali Nishadam* sounded too low even when the relative frequency of 243/128 was used. The note has generally to be coded as 'sa ni sa'. Similarly, *Pratimadhyamam* sounded too low even when the r.f. 64/45 was used, and the note had to be generally coded as 'pa ma pa'.

Importance of Note and Transit Durations

The most important point which emerged was that in the production of *gamakams* not only the extent of oscillation but the durations of the notes and the transitions were very important. Obviously, a computer analysis of live music would be of great value in this regard. The presently available tools for analysing music put more stress on the spectrum of the sound than on the actual frequencies, since in the West the frequencies are fixed. Fourier Transform techniques can introduce large errors when the frequency is to be ascertained over a very small time period, which is necessary for analysing the *gamakams* as the frequency keeps changing continuously. I am trying to develop an analytical tool for this purpose.

Music Generated from Notation

The second program developed by me, which enables the user to enter notation in *sa-ri-ga-ma* style, required an interface to convert the notation into frequencies and duration. This was not difficult and did not put too much strain on the computer's resources. But extending the notation system to represent *gamakam* quantitatively was found to be very difficult. The choice was between adopting a qualitative symbol for the *gamakam* (such as *kampitam*) and letting the computer generate the most common form of the *gamakam*, or to give the user total control on the flow of frequencies and durations. The latter option was taken to make the program more useful for a wide range of users. However, it also introduced considerable difficulties for the user.

Some minor changes were made in the notation system to avoid creating a new set of fonts and key correlations. Upper- and lower-case letters were used to indicate the octave instead of putting dots over and below the notes. To denote higher *kalams* (double or quadruple tempos), brackets were used instead of lines drawn over the notes. In fact, for accurate production of *gamakams*, sometimes even triple brackets — the equivalent of three lines drawn over the notation — were required.

Three major additions were made to make the notation more accurate, which is needed for the computer:

(a) The symbol '-' was introduced to separate groups of notes into phrases. At present, books use the symbols ', ' and ';' to indicate both silences of one or two notes' duration and to prolong the previous note. The hyphen symbol '-' makes it easy to indicate whether a silence is indicated, or a prolongation of the previous note.

(b) The symbols '\ ' and '/ ' were introduced to precisely define jaru, i.e., to indicate the transit durations.

(c) The symbols '<' and '>' were introduced to denote microtonal variations in the note pitches.

The other requirements of melam definition, choice of tempo, change of *melam* in the middle of a file (for *ragamalika*), change of tempo in the middle of a file (for *gati* change), etc., have been incorporated. Provision for a quick check of the total number of notes (after adjusting for half durations, quarter durations, etc.) is available. This check can be made for any selected part of the notation.

As for playing the notation, either all the notation or a selected part can be played. Continuous repeated playing of the notation by looping back is also possible, which may be useful for practice. A wide range of *adharasrutis* from .5 kattai to 5.5 kattai, i.e., a semitone below C up to a semitone above G, is available, and a choice of instrument (flute or Veena) is provided.

The programme is considered useful for copying notation from books and playing it. But for this to be really useful, *gamakams* have to be added by splitting notes by showing *anuswarams* explicitly and adjusting the durations of the notes and transitions. To make this easier, the program also provides readymade notations for some common ragams, which can be copied and pasted. It is also noticed that in the books silences are not explicitly noted, except where the composition starts after the beginning of an *avartam*, when a comma, or a semicolon, or their combinations are used. Elsewhere, these symbols are used both to prolong a note and to indicate gaps. But, for the computer to play the notation accurately, the user has to work out the actual note's duration and the silence that follows and put a hyphen symbol ('-') in between. Again, for a realistic reproduction of a *Kriti*'s words, silences are to be provided when joint consonants appear. For instance, when the word '*bhakta*' is pronounced, there is a very small gap in the continuity of voice before the consonant 'ta'. Normally, in writing notation, these finer aspects are not incorporated as they are played intuitively on instruments.

The gap between what is written down in the existing system of notation and what is actually sung is a major problem in trying to generate Carnatic music from notation on the computer. The major deficiencies of the notation system are (a) inadequate representation of *gamakams*; (b) lack of precision in describing silences; and (c) in the case of instruments, lack of indication of points of strumming (for the Veena) or bow change (for violin). Traditionally, these are left to be learnt intuitively.

There is another field in which a program of this kind could be useful — research in musical cognizance — since the required notes or phrases could be produced accurately — and no element of subjectivity on the part of the singer would come in. One could do research in specific areas of pitch recognition such as the thresholds of note duration for accurate pitch appreciation. Even raga recognition could be objectively researched.

In conclusion, it may be stated that though there are many problems in generating Carnatic

music synthetically with all its nuances, these are not insurmountable. Such programs would be useful in learning and teaching (especially practising lessons in-between tuitions with a teacher), and voice-training. Such programs can also help in enhancing the student's appreciation of the finer aspects of Carnatic music, and allow him or her to experiment. As mentioned earlier, the programs can also be useful for research in music, especially in the area of musical cognizance, where very little work has been done.